Please type a plus sign (+) inside this box → ☒ +

PTO/SB/05 (4/98)
Approved for use through 09/30/2000. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 C.F.R. § 1.53(b))

Attorney Docket No.	219.38696X00
First Inventor or Application Identifier	Bimal PODDAR
Title	See 1 in Addendum
Express Mail Label No.	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. ☒ * Fee Transmittal Form (e.g., PTO/SB/17)
(Submit an original and a duplicate for fee processing)
2. ☒ Specification [Total Pages 16]
(preferred arrangement set forth below)
- Descriptive title of the Invention
- Cross References to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to Microfiche Appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings (if filed)
- Detailed Description
- Claim(s)
- Abstract of the Disclosure
3. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 4]
4. Oath or Declaration [Total Pages 4]
a. ☒ Newly executed (original or copy)
b. □ Copy from a prior application (37 C.F.R. § 1.63(d))
(for continuation/divisional with Box 16 completed)
i. □ DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application, see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).

* NOTE FOR ITEMS 1 & 13: IN ORDER TO BE ENTITLED TO PAY SMALL ENTITY FEES, A SMALL ENTITY STATEMENT IS REQUIRED (37 C.F.R. § 1.27), EXCEPT IF ONE FILED IN A PRIOR APPLICATION IS RELIED UPON (37 C.F.R. § 1.28).

5. □ Microfiche Computer Program (Appendix)
6. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
a. □ Computer Readable Copy
b. □ Paper Copy (identical to computer copy)
c. □ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

7. ☒ Assignment Papers (cover sheet & document(s))
8. □ 37 C.F.R.§3.73(b) Statement (when there is an assignee) □ Power of Attorney
9. □ English Translation Document (if applicable)
10. ☒ Information Disclosure Statement (IDS)/PTO-1449 ☒ Copies of IDS Citations
11. □ Preliminary Amendment
12. ☒ Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
13. □ * Small Entity Statement(s) (PTO/SB/09-12) □ Statement filed in prior application Status still proper and desired
14. □ Certified Copy of Priority Document(s) (if foreign priority is claimed)
15. ☒ Other: Credit Card Payment Form

16. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:
□ Continuation □ Divisional □ Continuation-in-part (CIP) of prior application No: ________/__________________
Prior application information: Examiner ______________________ Group / Art Unit: ______________

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

17. CORRESPONDENCE ADDRESS

☒ Customer Number or Bar Code Label 020457 (Insert Customer No. or Attach bar code label here) or □ Correspondence address below

Name					
Address					
City		State		Zip Code	
Country		Telephone		Fax	

Name (Print/Type)	Henry M. Zykorie	Registration No. (Attorney/Agent)	27,477
Signature		Date	Sept. 29, 2000

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

Attachment to PTO/SB/05 (4/98) Utility Patent Application Transmittal

1. EFFICIENT DETECTION OF TEXTURE SHARING BETWEEN MULTIPLE CONTEXTS

PTO/SB/17 (12/99)
Approved for use through 09/30/2000 OMB 0651-0032
Patent and Trademark Office U S DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

FEE TRANSMITTAL for FY 2000

Patent fees are subject to annual revision.
Small Entity payments must be supported by a small entity statement, otherwise large entity fees must be paid See Forms PTO/SB/09-12 See 37 C.F.R §§ 1 27 and 1.28

TOTAL AMOUNT OF PAYMENT ($)808.00

Complete if Known	
Application Number	NEW
Filing Date	September 29, 2000
First Named Inventor	Bimal PODDAR
Examiner Name	
Group / Art Unit	
Attorney Docket No.	219.38696X00

METHOD OF PAYMENT (check one)

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any overpayments to.

Deposit Account Number: 01-2135

Deposit Account Name: Antonelli, Terry, Stout & Kraus, LLP

☒ Charge Any Additional Fee Required Under 37 CFR §§ 1.16 and 1 17

2. □ **Payment Enclosed:** □ Check □ Money Order ☒ Other

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	690	201	345	Utility filing fee	690.00
106	310	206	155	Design filing fee	
107	480	207	240	Plant filing fee	
108	690	208	345	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1) ($) 690.00

2. EXTRA CLAIM FEES

			Extra Claims		Fee from below		Fee Paid
Total Claims	16	-20** =	0	x	18	=	0
Independent Claims	4	- 3** =	1	x	78	=	78
Multiple Dependent						=	0

***or number previously paid, if greater; For Reissues, see below*

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	9	Claims in excess of 20
102	78	202	39	Independent claims in excess of 3
104	260	204	130	Multiple dependent claim, if not paid
109	78	209	39	** Reissue independent claims over original patent
110	18	210	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 78.00

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	0.00
127	50	227	25	Surcharge - late provisional filing fee or cover sheet.	0.00
139	130	139	130	Non-English specification	0.00
147	2,520	147	2,520	For filing a request for reexamination	0.00
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	0.00
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	0.00
115	110	215	55	Extension for reply within first month	0.00
116	380	216	190	Extension for reply within second month	0.00
117	870	217	435	Extension for reply within third month	0.00
118	1,360	218	680	Extension for reply within fourth month	0.00
128	1,850	228	925	Extension for reply within fifth month	0.00
119	300	219	150	Notice of Appeal	0.00
120	300	220	150	Filing a brief in support of an appeal	0.00
121	260	221	130	Request for oral hearing	0.00
138	1,510	138	1,510	Petition to institute a public use proceeding	0.00
140	110	240	55	Petition to revive - unavoidable	0.00
141	1,210	241	605	Petition to revive - unintentional	0.00
142	1,210	242	605	Utility issue fee (or reissue)	0.00
143	430	243	215	Design issue fee	0.00
144	580	244	290	Plant issue fee	0.00
122	130	122	130	Petitions to the Commissioner	0.00
123	50	123	50	Petitions related to provisional applications	0.00
126	240	126	240	Submission of Information Disclosure Stmt	0.00
581	40	581	40	Recording each patent assignment per property (times number of properties)	40.00
146	690	246	345	Filing a submission after final rejection (37 CFR § 1 129(a))	0.00
149	690	249	345	For each additional invention to be examined (37 CFR § 1 129(b))	0.00
				Other fee (specify)________________	0.00
				Other fee (specify)________________	0.00

* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 40.00

SUBMITTED BY				Complete *(if applicable)*	
Name (Print/Type)	Henry M. Zykorie	*Registration No (Attorney/Agent)*	27,477	*Telephone*	703-312-6600
Signature				*Date*	Sept. 29, 2000

WARNING:

Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Burden Hour Statement. This form is estimated to take 0 2 hours to complete Time will vary depending upon the needs of the individual case Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231 DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS SEND TO· Assistant Commissioner for Patents, Washington, DC 20231.

219.38696X00
P9336

UNITED STATES PATENT APPLICATION FOR:

EFFICIENT DETECTION OF TEXTURE SHARING BETWEEN MULTIPLE CONTEXTS

Inventor:

Bimal PODDAR

Prepared by:

Antonelli, Terry, Stout & Kraus, LLP
1300 North 17th Street, Suite 1800
Arlington, VA 22209
Telephone: 703/312-6600
Telecopier: 703/312-6666

219.38696X00
P936

EFFICIENT DETECTION OF TEXTURE SHARING BETWEEN MULTIPLE CONTEXTS

FIELD

The present invention relates to texture sharing in an openGL API (Application Programming Interface) or other graphics programming interface. More particularly, the present invention relates to a technique for effecting an efficient detection of texture sharing between multiple contexts.

BACKGROUND

Graphics programming interfaces, such as the openGL API, are being rapidly developed to keep up with their great demand for use in graphics displays in computer applications. U.S. Patent No. 5,818, 469, to John Joseph Lawless et al. discloses an example of a graphics programming interface.

Figure 1, which illustrates a schematic representation of a graphics architecture shows a graphics application 101 which is typically running on a workstation or other computer system. A graphics interface 103 receives data from the graphics application 101 and processes the data prior to forwarding it to hardware 115 using a plurality of threads. A thread is a predefined program segment and is operable to effect the accomplishment of a specified individual graphics task such as rasterizing or rendering. A plurality of threads 107 and 109 to be used for rendering are fed from the graphic interface 103 to the hardware 115. Each thread maintains its own local graphics context containing the attribute state. Threads 107 and 109 may include related graphics contexts 108 and110, respectively associated therewith.

Figure 2, which illustrates a simplified block diagram of a computer system, shows an exemplary system 200 in which the graphics subsystem 217 corresponds to the hardware 115 of Figure 1. A central processing unit (CPU) 201 is connected to a central bus 203. A memory subsystem 205 and cache memory 207 are also connected to the bus 203 as is a storage block 215 which may include one or more storage devices such as floppy disk drives, hard drives, etc. A display device 219 is connected to the graphics subsystem 217. An input interface device 209 connects a keyboard 211 and a mouse 213 to the bus 203. The bus 203 may be extended 221 to be connected to other systems and/or devices.

The graphics subsystem 217 typically includes an internal graphics processor as well as a frame buffer memory for use in connection with the display device. For example, the graphics subsystem 217 generally includes rasterization hardware as well as other specific graphics engines. Instructions for performing graphics interfacing processes may be executed by the processor 201 and/or a separate graphics processor disposed within the graphics subsystem 217. To such instructions may be embodied within or stored in any one of or a combination of storage devices and/or memory devices including RAM memory within the memory subsystem 205, any of the storage elements of the storage block 215 or any portable storage device, such as a floppy disk or CD.

The openGL API provides for textures to be shared by multiple rendering contexts. In addition, each context may have multiple texture units that use the same texture. For certain operations with a texture, it is important to distinguish between multiple contexts using a texture or a single context using a texture in multiple units.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and a better understanding of the present invention will become apparent from the following detailed description of example embodiments and the claims when read in connection with the accompanying drawings, all forming a part of the disclosure of this invention. While the foregoing and following written and illustrated disclosure focuses on disclosing example embodiments of the invention, it should be clearly understood that the same is by way of illustration and example only and the invention is not limited thereto. The spirit and scope of the present invention is limited only by the terms of the appended claims.

The following represents brief descriptions of the drawings, wherein:

Figure 1 illustrates a schematic representation of a graphics architecture arrangement.

Figure 2 illustrates a simplified block diagram of a computer system.

Figure 3 illustrates an example used for explaining the technique in accordance with the present invention.

Figure 4 illustrates a flowchart used for explaining the technique in accordance with the present invention.

DETAILED DESCRIPTION

Before beginning a detailed description of the subject invention, mention of the following is in order. When appropriate, like reference numerals and characters may be used to designate identical, corresponding, or similar components in differing drawing figures. Furthermore, in the detailed description to follow, example sizes/models/values/ranges may be given, although the present invention is not limited thereto. In addition, various connections and other elements may not be shown within the drawing figures for simplicity of illustration and discussion and so as not to obscure the invention.

The openGL API, as with other graphics APIs, provides for a rendering context. The rendering context encapsulates all of the states necessary to render triangles with attributes such as color, fog, etc. The context information also includes texture images that may consume large amounts of memory. OpenGL allows an application to have multiple contexts so that the separate context can be bound to multiple threads to allow simultaneous rendering by multiple threads. However, one context can be bound to only one thread. Since the texture images are fairly large, openGL allows for sharing of textures by multiple contexts. When things are shared between multiple threads, it is a common practice to include mutual exclusion locks to prevent a second thread from using the same texture object while the first thread is using the texture object. However, in order to avoid a stall of any thread for a substantial time period, the mutual exclusion lock may be used only for a short time period to set a read lock flag in the texture and then release the mutual exclusion lock. The read lock flag could be implemented by simply updating a reference counter of the texture. This allows both the threads to simultaneously render using the same texture.

Due to recent advances in graphics hardware, openGL, for example, allows for each context to have multiple texture units. The result of a blending of a polygon color and a lookup from one texture unit are passed onto the next texture unit. A single texture may be assigned to multiple units in a multi-texture pipeline.

An application may request the driver to change the texture image or any of its attributes. In order to safely change the texture image on one thread, it is important to know whether the texture is being used by only this thread/context or is being used by multiple threads/contexts. If each openGL context supported only one texture unit and if a texture had a reference count of more than 1, then there would be an implication that the texture is being used by multiple contexts and could not be safely modified. However, due to the presence of multiple texture units, a reference count does not provide all the information. For example, a reference count of 2 may indicate one of two alternatives, namely, either a texture is being used by a single texture unit on multiple contexts or is being used by multiple texture units on a single context. If the texture was being used by a single context on multiple texture units, then the texture could be safely modified but if the texture was used by multiple contexts then the texture could not safely be modified.

In order to determine whether a texture is being used by only a single context, it is possible to compare the texture with all of the texture units on a context and determine a usage count for a context. If the usage count matched the reference count for the texture, it would imply that the texture was being used only on this context. The determination of the usage count requires one compare operation and potentially one addition operation per texture unit. With the rapid improvements in graphics hardware, there is a reasonable expectation that the number of texture units may increase to 32 texture units. Thus, this method of determining the usage of a texture becomes rather expensive when applications require fast state changes for optimal performance.

In the present invention, when contexts request the sharing of a set of textures, each context is provided with its own unique identifier (ID). This unique identifier is a single bit in an integer word. When a context starts using a texture, it bitwise ORs its unique context ID into a texture usage mask. Each texture has its own texture usage mask. This texture usage mask tracks the identifiers of all of the contexts using the corresponding texture. If only one bit is turned on in the texture usage mask, then only one context is using the texture. Accordingly, to determine whether a texture is being used by another context, only the following simple test need be performed.

If the texture usage mask ANDed with the inverted context ID is equal to 0, then the texture is not being used by another context. If the texture usage mask ANDed with the inverted context ID is not equal to 0, then the texture is being used by another context.

Figure 3 illustrates an example used for explaining the technique in accordance with the present invention. As illustrated in Figure 3, there are two contexts, Context 1 and Context 2, each context having 4 texture units. Texture A is being used by both Context 1 and Context 2 while Texture B is being used in multiple units only on Context 1. As can be seen from Figure 3, Texture A has a reference count of 2. Furthermore, the texture usage mask for the texture, which is obtained by bitwise ORing of the two context IDs, indicates that both contexts are using the texture. On the other hand, Texture B, which also has a reference count of 2, only has the context ID bit for Context 1 set in its texture usage mask, thereby indicating that only a single context is using the texture. Thus, while the share status of a texture cannot be determined by the reference count of a texture, the texture usage mask for a texture clearly indicates whether the texture is being used by a single context or by multiple contexts. With the present invention, the determination of texture usage for multiple contexts can be effected using only one invert operation and one compare operation. In fact, the invert operation may be eliminated by maintaining a precomputed inverted context ID for each context.

Figure 4 illustrates a flowchart used for explaining the technique in accordance with a present invention. Upon starting the technique for detecting the sharing of a texture between multiple contexts, in block 410, the texture usage mask of a subject texture is obtained.

In block 420, the context ID of a subject context is obtained and in block 430, a first logic operation is performed to produce a resultant value. As noted above, for example, the first logic operation might be the ANDing of the texture usage mask of the subject texture with an inverted context ID of the subject context to produce the resultant value.

In block 440, a determination is made as to whether the resultant value is equal to a first or second predetermined value. If the resultant value is equal to the first predetermined value, then the subject texture is not being shared while if the resultant value is equal to the second predetermined value, then the subject texture is being shared. As noted above, for example, the first predetermined value may be equal to 0 while the second predetermined value may be equal to any value other than 0.

When a context stops using a texture, it's context ID bit in the corresponding texture usage mask must be reset. This is performed as follows:

1. For the texture which is no longer going to be used, clear the context ID bit in its texture usage mask.

2. Attach a new texture to the texture unit.

3. Loop over all of the texture units in a context.

For each texture, set the context ID bit for this context in its corresponding texture usage mask.

The above-noted operation performs as many bitwise OR operations as the number of texture units. Note that multiple compare operations associated with the reference count scheme have been eliminated. In deeply pipelined hardware CPU implementations, compare operations may be computationally very expensive. In addition, the operation is performed only when a unit stops using a texture. The normal mode for a texture unit is to attach a texture, perform multiple operations on the texture, and then stop using the texture. Thus, with the present invention, operations that are proportional to the number of texture units are performed only once when a texture is detached from a texture unit.

In addition to modifying a texture, an openGL engine, for example, is supposed to virtualize textures. That is, if a system does not have enough hardware texture memory, then the driver is supposed to swap out some textures to the system memory and page in other textures from the system memory to the hardware texture memory. The texture manager, when executing one thread, must insure that it does not page out a texture that is actively being used by another texture unit. This can easily be determined by using the unique context IDs and the texture usage masks of the present invention rather than using a reference count which does not provide a complete answer.

Compared to existing techniques, the present invention reduces the number of operations from being on the order of the number of texture units to just one operation most of the time. Even when the number of operations is proportional to the number of texture units, the technique in accordance with the present invention is faster due to the elimination of costly compare operations.

This concludes the description of the example embodiments. Although the present invention has been described with reference to illustrative embodiments, it is to be understood that numerous other modifications and embodiments can be devised by those skilled in the art that will fall within the spirit and scope of the principles of this invention. More particularly, reasonable variations and modifications are possible in the component parts and/or arrangements of the subject combination arrangement within the scope of the foregoing disclosure, the drawings, and the appended claims without departing from the spirit of the invention. In addition to variations and modifications in the component parts and/or arrangements, alternative uses will be apparent to those skilled in the art.

What is claimed is:

CLAIMS

1. A method of detecting texture sharing between multiple contexts having unique context ID's:

obtaining a texture usage mask of a subject texture;

obtaining an inverted context ID of a subject context;

ANDing the texture usage mask of the subject texture with the inverted context ID of the subject context to produce a resultant value; and

detecting that the subject texture is not being shared by another context with the subject context upon the resultant value being equal to 0 and detecting that the subject texture is being shared by another context with the subject context upon the resultant value not being equal to 0.

2. The method of claim **1**, further comprising:

revising the texture usage mask of a subject texture prior to the subject texture being used by another context by bitwise ORing the texture usage mask with a context ID of the another context to produce a resultant new texture usage mask for the subject texture.

3. The method of claim **1**, further comprising:

revising the texture usage mask of a subject texture upon the subject texture no longer being used by a particular context by deleting a context ID of the particular context from the texture usage mask to produce a resultant new texture usage mask for the subject texture.

4. The method of claim **2**, further comprising:

revising the texture usage mask of a subject texture upon the subject texture no longer being used by a particular context by deleting a context ID of the particular context from the texture usage mask to produce a resultant new texture usage mask for the subject texture.

5. A method of detecting texture sharing between multiple contexts having unique context ID's:

obtaining a texture usage mask of a subject texture;

obtaining a context ID of a subject context;

performing a first logic operation with the texture usage mask of the subject texture and the context ID of the subject context to produce a resultant value; and

detecting that the subject texture is not being shared by another context with the subject context upon the resultant value being equal to a first predetermined value and detecting that the subject texture is being shared by another context upon the resultant value being equal to a second predetermined value which is different from the first predetermined value.

6. The method of claim **5**, further comprising:

revising the texture usage mask of a subject texture prior to the subject texture being used by another context by performing a second logic operation with the texture usage mask and a context ID of the another context to produce a resultant new texture usage mask for the subject texture.

7. The method of claim **5**, further comprising:

revising the texture usage mask of a subject texture upon the subject texture no longer being used by a particular context by performing a third logic operation with the texture usage mask and a context ID of the particular context to produce a resultant new texture usage mask for the subject texture.

8. The method of claim **7**, further comprising:

revising the texture usage mask of a subject texture upon the subject texture no longer being used by a particular context by performing a third logic operation with the texture usage mask and a context ID of the particular context to produce a resultant new texture usage mask for the subject texture.

9. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method of detecting texture sharing between multiple contexts having unique context ID's, the method comprising:

obtaining a texture usage mask of a subject texture;

obtaining an inverted context ID of a subject context;

ANDing the texture usage mask of the subject texture with the inverted context ID of the subject context to produce a resultant value; and

detecting that the subject texture is not being shared by another context with the subject context upon the resultant value being equal to 0 and detecting that the subject texture is being shared by another context with the subject context upon the resultant value not being equal to 0.

10. The program storage device of claim **9**, the method further comprising:

revising the texture usage mask of a subject texture prior to the subject texture being used by another context by bitwise ORing the texture usage mask with a context ID of the another context to produce a resultant new texture usage mask for the subject texture.

11. The program storage device of claim **9**, the method further comprising:

revising the texture usage mask of a subject texture upon the subject texture no longer being used by a particular context by deleting a context ID of the particular context from the texture usage mask to produce a resultant new texture usage mask for the subject texture.

12. The program storage device of claim **11**, the method further comprising:

revising the texture usage mask of a subject texture upon the subject texture no longer being used by a particular context by deleting a context ID of the particular context from the texture usage mask to produce a resultant new texture usage mask for the subject texture.

13. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method of detecting texture sharing between multiple contexts having unique context ID's, the method comprising:

obtaining a texture usage mask of a subject texture;

obtaining a context ID of a subject context;

performing a first logic operation with the texture usage mask of the subject texture and the context ID of the subject context to produce a resultant value; and

detecting that the subject texture is not being shared by another context with the subject context upon the resultant value being equal to a first predetermined value and detecting that the subject texture is being shared by another context upon the resultant value being equal to a second predetermined value which is different from the first predetermined value.

14. The program storage device of claim **13**, the method further comprising:

revising the texture usage mask of a subject texture prior to the subject texture being used by another context by performing a second logic operation with the texture usage mask and a context ID of the another context to produce a resultant new texture usage mask for the subject texture.

15. The program storage device of claim **13**, the method further comprising:

revising the texture usage mask of a subject texture upon the subject texture no longer being used by a particular context by performing a third logic operation with the texture usage mask and a context ID of the particular context to produce a resultant new texture usage mask for the subject texture.

16. The program storage device of claim **15**, the method further comprising:

revising the texture usage mask of a subject texture upon the subject texture no longer being used by a particular context by performing a third logic operation with the texture usage mask and a context ID of the particular context to produce a resultant new texture usage mask for the subject texture.

ABSTRACT

A technique for detecting texture sharing between multiple contexts having unique ID's includes obtaining a texture usage mask of a subject texture and a context ID of a subject context. A first logic operation with the texture usage mask of the subject texture and the context ID of the subject context is performed to produce a resultant value. The subject texture is determined not to be shared by another context with the subject context upon the resultant value being equal to a first predetermined value. The subject texture is determined to be shared by another context with the subject context upon the resultant value being equal to a second predetermined value which is different from the first predetermined value. The texture usage mask of a subject texture may be revised prior to the subject texture being used by another context by performing a second logic operation with the texture usage mask and a context ID of another context to produce a resultant new texture usage mask for the subject texture. The texture usage mask of a subject texture may be revised upon the subject texture no longer being used by a particular context by performing a third logic operation with the texture usage mask and a context ID of the particular context to produce a resultant new texture usage mask for the subject texture.

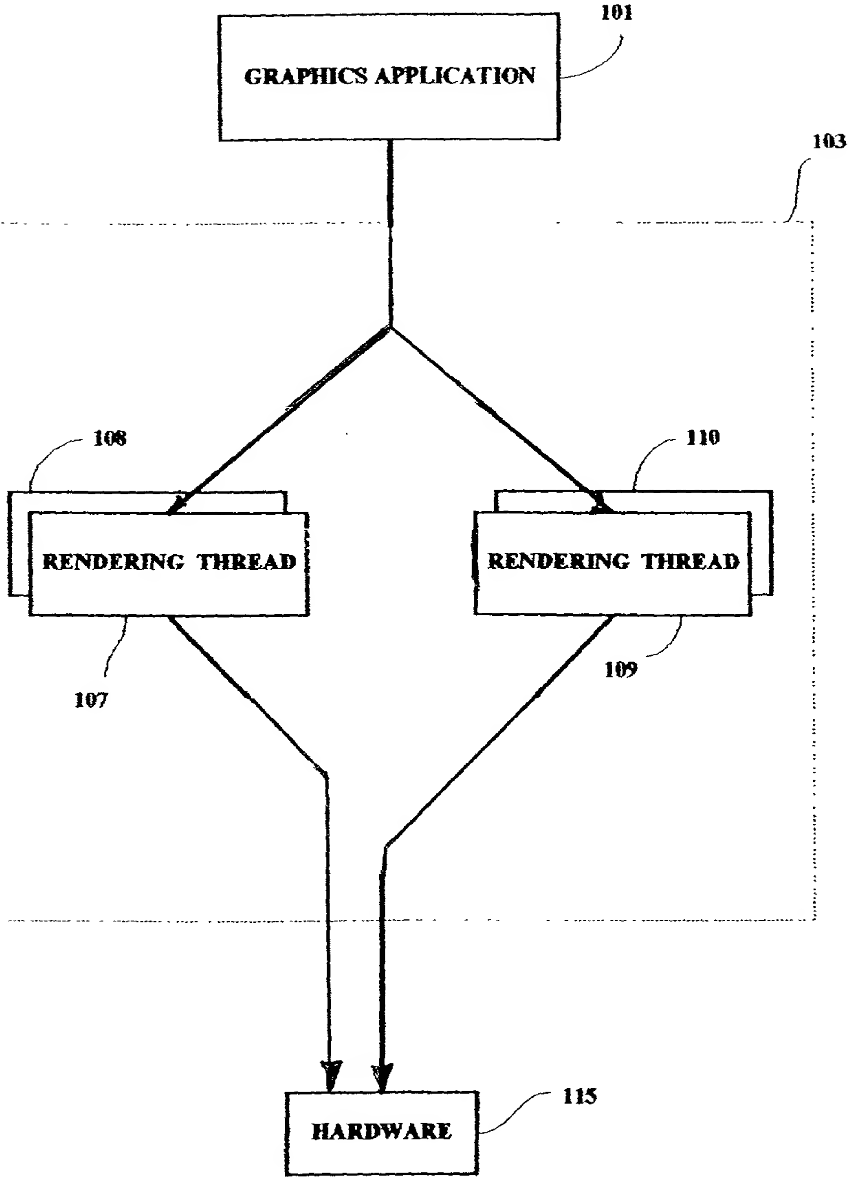
101
GRAPHICS APPLICATION
103
108
110
RENDERING THREAD
RENDERING THREAD
107
109
HARDWARE
115

FIG. 1

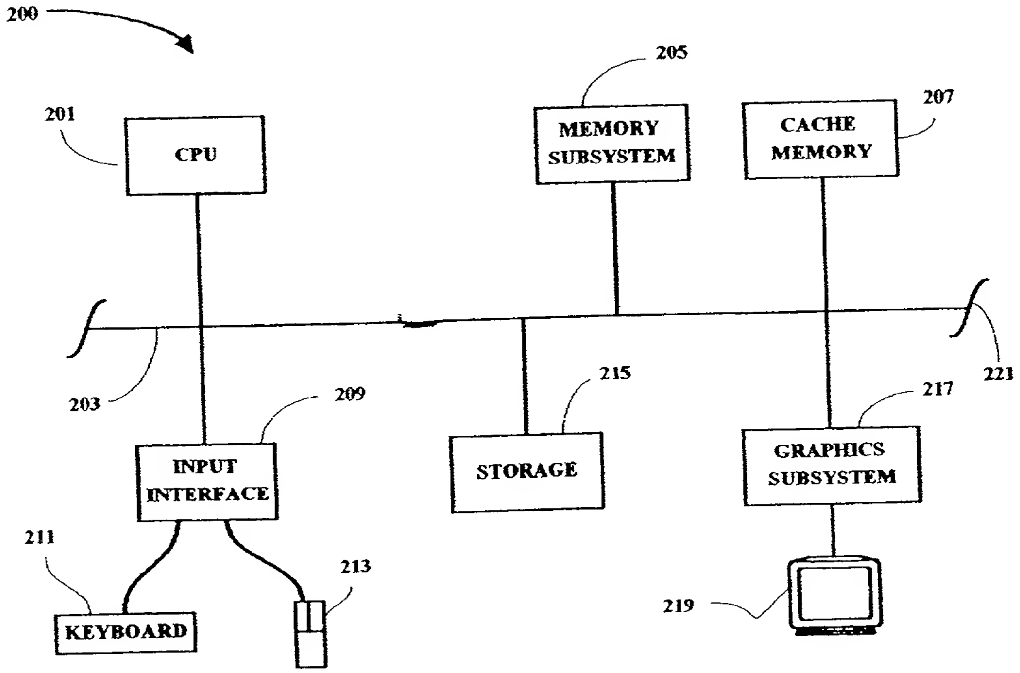
200
201
CPU
205
MEMORY SUBSYSTEM
CACHE MEMORY
207
203
209
INPUT INTERFACE
215
STORAGE
217
GRAPHICS SUBSYSTEM
221
211
KEYBOARD
213
219

FIG. 2

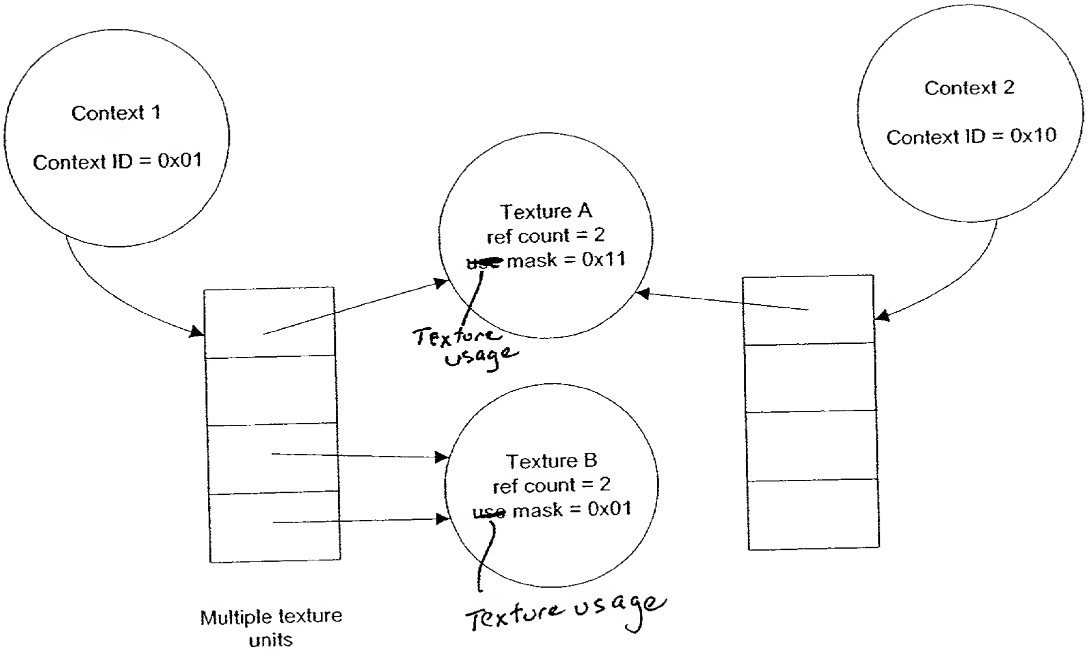
Context 1
Context ID = 0x01
Texture A
ref count = 2
~~use~~ mask = 0x11
Texture usage
Context 2
Context ID = 0x10
Texture B
ref count = 2
~~use~~ mask = 0x01
Texture usage
Multiple texture units

FIG. 3

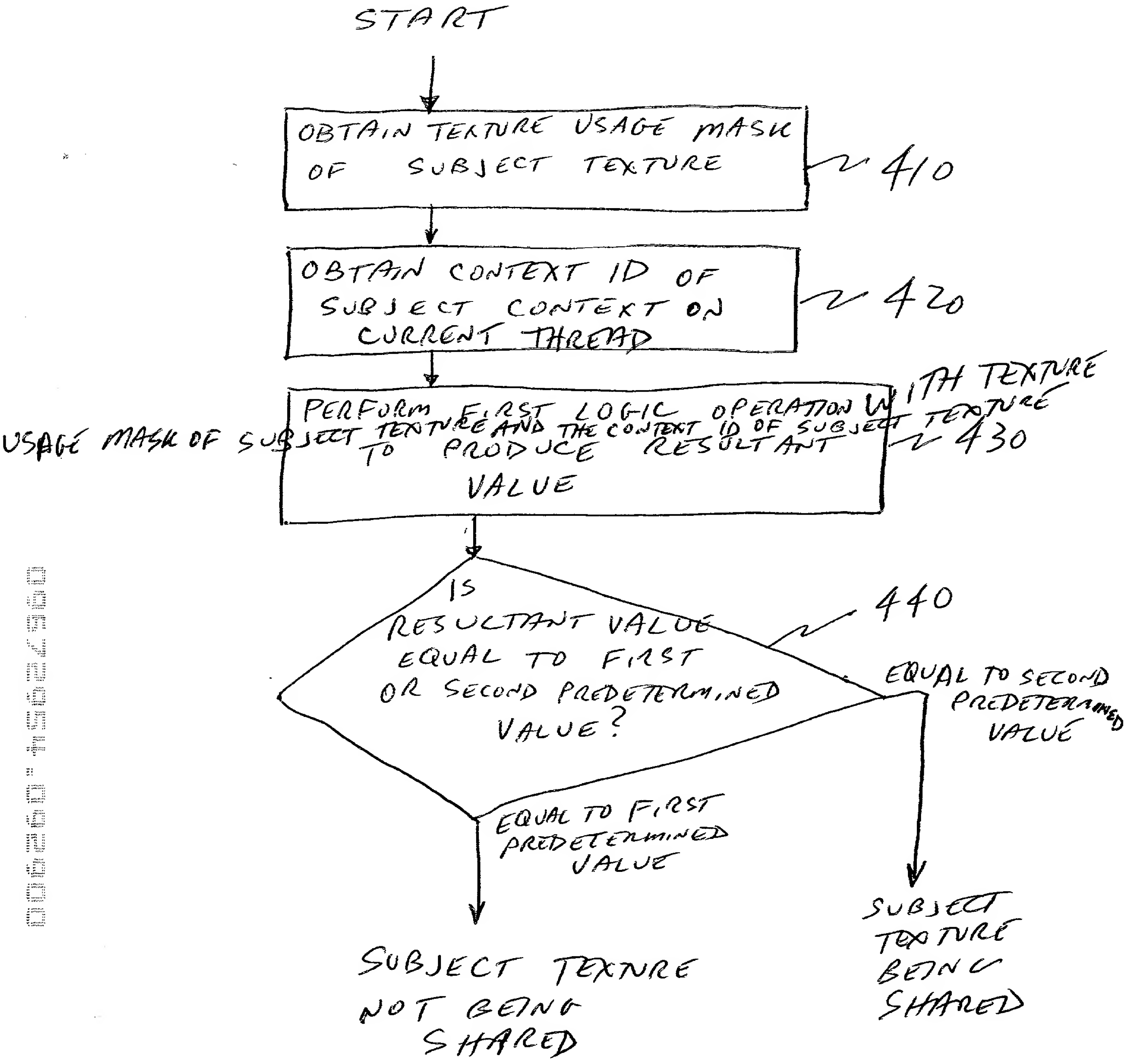
START
OBTAIN TEXTURE USAGE MASK OF SUBJECT TEXTURE
410
OBTAIN CONTEXT ID OF SUBJECT CONTEXT ON CURRENT THREAD
420
PERFORM FIRST LOGIC OPERATION WITH TEXTURE USAGE MASK OF SUBJECT TEXTURE AND THE CONTEXT ID OF SUBJECT TEXTURE TO PRODUCE RESULTANT VALUE
430
IS RESULTANT VALUE EQUAL TO FIRST OR SECOND PREDETERMINED VALUE?
440
EQUAL TO SECOND PREDETERMINED VALUE
EQUAL TO FIRST PREDETERMINED VALUE
SUBJECT TEXNRE NOT BEING SHARED
SUBJECT TEXTURE BEING SHARED

FIG. 4

Attorney's Docket No.: 219.38696X00(ATSK) PATENT
Intel No. P9336

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION
(FOR INTEL CORPORATION PATENT APPLICATIONS)

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled EFFICIENT DETECTION OF TEXTURE SHARING BETWEEN MULTIPLE CONTEXTS

the specification of which
x is attached hereto.
__ was filed on __________________ as
United States Application Number ____________
or PCT International Application Number __________
and was amended on __________________.
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above. I do not know and do not believe that the claimed invention was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (for a utility patent application) or six months (for a design patent application) prior to this application.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d), of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No
(Number)	(Country)	(Day/Month/Year Filed)	Yes	No

I hereby claim the benefit under title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below

(Application Number) Filing Date

(Application Number) Filing Date

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Number) Filing Date (Status -- patented, pending, abandoned)

(Application Number) Filing Date (Status -- patented, pending, abandoned)

I hereby appoint: Donald R. Antonelli, Reg. No. 20,296; David T. Terry, Reg. No. 20,178; Melvin Kraus, Reg. No. 22,466; William I. Solomon, Reg. No. 28,565; Gregory E. Montone, Reg. No. 28,141; Ronald J. Shore, Reg. No. 28,577; Donald E. Stout, Reg. No. 26,422; Alan E. Schiavelli, Reg. No. 32,087; James N. Dresser, Reg. No. 22,973; Carl I. Brundidge, Reg. No. 29,621; Paul J. Skwierawski, Reg. No. 32,173, my attorneys; of ANTONELLI, TERRY, STOUT & KRAUS, LLP with offices located at 1300 North Seventeenth Street, Suite 1800, Arlington, Virginia 22209, telephone: (703) 312-6600, fax: (703) 312-6666; and Alan K. Aldous, Reg. No. 31,905; Robert D. Anderson, Reg. No. 33,826; Joseph R. Bond, Reg. No. 36,458; R. Edward Brake, Reg. No. 37,784; Richard C. Calderwood, Reg. No. 35,468; Jeffrey S. Draeger, Reg. No. 41,000; Cynthia Thomas Faatz, Reg No. 39,973; Sean Fitzgerald, Reg. No. 32,027; Seth Z. Kalson, Reg. No. 40,670; David J. Kaplan, Reg. No. 41,105; Leo V. Novakoski, Reg. No. 37,198; Naomi Obinata, Reg. No. 39,320; Thomas C. Reynolds, Reg. No. 32,488; Steven P. Skabrat, Reg. No. 36,279; Howard A. Skaist, Reg. No. 36,008; Steven C. Stewart, Reg. No. 33,555; Raymond J. Werner, Reg. No. 34,752; and Charles K. Young, Reg. No. 39,435; my patent attorneys, and Calvin E. Wells, Reg. No. P43,256; and Alexander Ulysses Witkowski, Reg. No. P43,280; my patent agents, of INTEL CORPORATION; with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

Send all correspondence to:

ANTONELLI, TERRY, STOUT & KRAUS, LLP
1300 North 17th Street, Suite 1800
Arlington, VA 22209

Direct all telephone calls and faxes to:

TEL: (703) 312-6600
FAX: (703) 312-6666

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor Bimal PODDAR

Inventor's Signature Bimal Poddar Date 9/28/2000

Residence El Dorado Hills, California Citizenship ~~U.S.A.~~ INDIA
(City, State) (Country)

Post Office Address 4710 Freno Way
El Dorado Hills, CA 95762

Full Name of Second/Joint Inventor ______

Inventor's Signature ______ Date ______

Residence ______ Citizenship ______
(City, State) (Country)

Post Office Address ______

Full Name of Third/Joint Inventor ______

Inventor's Signature ______ Date ______

Residence ______ Citizenship ______
(City, State) (Country)

Post Office Address ______

Full Name of Fourth/Joint Inventor ______

Inventor's Signature ______ Date ______

Residence ______ Citizenship ______
(City, State) (Country)

Post Office Address ______

Title 37, Code of Federal Regulations, Section 1.56
Duty to Disclose Information Material to Patentability

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclosure information exists with respect to each pending claim until the claim is cancelled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is cancelled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclosure all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by ¤¤1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) Prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) The closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made or record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application;

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.